

# Service Oriented MAS: An open architecture

## (Extended Abstract)

C. Carrascosa, A. Giret, V. Julian, M. Rebollo, E. Argente, V. Botti  
Departamento de Sistemas Informaticos y Computacion  
Universidad Politecnica de Valencia  
Camino de Vera s/n, 46022. Valencia - Spain  
{carrasco;agiret;vinglada;mrebollo;eargente;vbotti}@dsic.upv.es

### ABSTRACT

This paper defines THOMAS, an open architecture and computational model for large-scale open multi-agent systems based on a service-oriented approach, specifically addressed for the design of virtual organizations.

### Categories and Subject Descriptors

D.2 [Software Engineering]: Software Architectures

### General Terms

Design

### Keywords

Agent Architecture, Service Oriented Multi-agent Systems, Open Systems, Social Structures

## 1. INTRODUCTION

Over recent years, several works have focused on solving the problem of integrating the multi-agent system and the service-oriented computing paradigms in order to model autonomous and heterogeneous computational entities in dynamic, open environments. Two approaches can be identified: (i) direct integration of web services and agents using message exchange ([4]) and (ii) considering agents as matchmakers for service discovering and composition ([6]).

A key problem for open MAS development is the existence of real agent platforms that support organizational concepts. Many agent platforms and agent architectures have been proposed, some of them focused on organizational concepts [1], but the majority are lacking in the management of virtual organizations for dynamic, open and large-scale environments. Designers must implement nearly all of the organizational features by themselves, namely organization representation and descriptions, control mechanisms, AMS and DF extensions, monitoring, organization modeling support and organizational API.

In this paper, it is defined a new open multi-agent system architecture, named THOMAS (MeTHods, Techniques and

**Cite as:** Service Oriented MAS: An open architecture, (Extended Abstract), C. Carrascosa, A. Giret, V. Julian, M. Rebollo, E. Argente, V. Botti, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 1291–1292  
Copyright © 2009, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org), All rights reserved.

Tools for *Open Multi-Agent Systems*), consisting of a related set of modules that are suitable for the development of systems applied in open environments in which heterogeneous entities (agents and services) interact. The proposed solution tries to communicate agents and web services in an independent way, thus raising a total integration of both multi-agent and service-oriented technologies. Therefore, agents can offer and invoke services in a transparent way to other agents or entities, as well as external entities can interact with THOMAS agents through the use of the offered services. Thus, THOMAS not only deals with the management of the virtual organization, such as in S-Moise+ [5] and Ameli (EI platform) [3], but it also includes a prototype platform fully addressed for open systems which has all organizational features in mind, trying to obtain a framework wholly independent of any internal agent platform.

## 2. THOMAS ARCHITECTURE

THOMAS architecture basically consists of a set of modular services. Though THOMAS feeds initially on the FIPA architecture, it expands its capabilities to deal with organizations, and to boost up its services abilities. In this way, a new module in charge of managing organizations has been introduced into the architecture, along with a redefinition of the FIPA *Directory Facilitator* that is able to deal with services in a more elaborated way, following *Service Oriented Architectures* guidelines. The main components of THOMAS are the following: *Service Facilitator* (SF), which offers simple and complex services to the active agents and organizations; *Organization Management System* (OMS), responsible of the management of the organizations and their entities; and *Platform Kernel* (PK) which maintains basic management services for an agent platform and represents any FIPA compliant platform.

### 2.1 Service Facilitator

The Service Facilitator (SF) is a mechanism and support by which organizations and agents can offer and discover services. The SF provides a place in which the autonomous entities can register service descriptions as directory entries.

The SF acts as a gateway to access the THOMAS platform. It manages this access transparently, by means of security techniques and access rights management. The SF can find services searching for a given service profile or searching by the goals that can be fulfilled when executing the service. This is done using the matchmaking and service composition mechanisms that are provided by the SF. The

SF also acts as a yellow pages manager and in this way it can find which entities provide a given service.

A service can be supplied by more than one provider in the system. Thus, a service has an associated list of providers, who can offer exact copies of the service, i.e. they share a common implementation of the service; or they may only share the service interface and each provider may implement it in a different way. This is easily achieved in THOMAS since the service profile is separated from the service process.

The SF supplies a set of standard services (named meta-services) to manage the services provided by organizations or individual agents. These meta-services can also be used by the rest of THOMAS components to advertise their own services. SF meta-services are classified in: *Registration* (for adding, modifying and removing services from the SF directory); *Affordability* (for managing the association between providers and their services) and *Discovery* (for searching and composing services as an answer to user requirements).

## 2.2 OMS

The *Organization Management System (OMS)* is in charge of the organization life-cycle management, including specification and administration of both the structural components of the organization (roles, units and norms) and its execution components (participant agents and roles they play).

Organizations are structured by means of *organizational units (OUs)*, which represent groups of entities (agents or other units), that are related in order to pursue a common goal. OUs can also be seen as virtual meeting points because agents can dynamically enter and leave them by means of adopting (or leaving) roles inside. Roles represent all required functionality needed in order to achieve the unit goal. They might also have associated norms for controlling role actions. Agents can dynamically adopt roles inside units, so a control for role adoption is needed. Finally, services represent some functionality that agents offer to other entities, independently of the concrete agent that makes use of them.

The OMS keeps record on which are the Organizational Units of the system, the roles defined in each unit and their attributes, the entities participating inside each OU and the roles that they enact through time. Moreover, the OMS also stores which are the norms defined in the system.

The OMS offers a set of services for organization life-cycle management, classified in: *Structural* (for modifying the structural/normative organization specification); *Informative* (for informing of the current state of the organization); and *Dynamic* (for managing dynamic entry/exit of agents and role adoption). Thus, it includes services for creating new organizations, admitting new members within those organizations and member resigning. By means of the publication of the *structural services*, the OMS allows the modification of some aspects related to the organization structure, functionality or normativity at execution time.

## 3. THOMAS FRAMEWORK

The THOMAS Framework is composed of both OMS and SF modules. Its purpose is to obtain a product wholly independent of any internal agent platform, thus fully addressed for open systems. This framework is based upon the idea that no internal agent exists, and the architecture services are offered as web services. In this way, only the OMS and the SF are composing such framework (avoiding the use of the PK due to the lack of internal agents to control).

Therefore, the THOMAS framework allows any agent to create a virtual organization with the structure and norms he wants, along with the demanding and offering services that he needs. The framework is in charge of the management of this organization structure, norms and life cycle, on one hand. On the other hand, it also controls the visibility of the offered and demanded services and the fulfillment of the conditions to use them. But, as it is fully addressed to open systems, the framework does not control the involved agents lifecycle, being all of them external to the framework.

## 4. CONCLUSIONS

The main contribution of this paper is the definition of an open architecture for large scale open multi-agent systems based on a service-oriented approach. There exists a current research interest in the integration of agents and services, agents being complex entities that can handle the problem of service discovery and composition in dynamic and changing open environments. Current agent architectures and platforms must integrate these concepts to allow designers to employ higher abstractions when modeling and implementing these complex systems. All of these concerns are gathered in the THOMAS proposal.

A detailed description of the THOMAS architecture and the first version of this framework (v0.1) is available in the THOMAS homepage<sup>1</sup>. The v0.1 framework implements the whole set of services described in the abstract architecture, with a basic support for norm management. This version has been used to check the feasibility of this approach with several examples using JADE and SPADE [2] agents.

## 5. ACKNOWLEDGMENTS

This work has been partially funded by TIN2008-04446, TIN2006-14630-C03-01, PROMETEO/2008/051 and GVPRE/2008/070 projects and CONSOLIDER-INGENIO 2010 under grant CSD2007-00022.

## 6. REFERENCES

- [1] E. Argente, A. Giret, S. Valero, V. Julian, and V. Botti. Survey of MAS Methods and Platforms focusing on organizational concepts. In *Proc. CCIA '04*, pages 309–316, 2004.
- [2] M. Escrivà, J. Palanca, G. Aranda, A. García-Fornes, V. Julian, and V. Botti. A jabber-based multi-agent system platform. In *Proc. of AAMAS06*, pages 1282–1284, 2006.
- [3] M. Esteva, J. Rodríguez-Aguilar, C. Sierra, J. Arcos, and P. García. *On the Formal Specification of Electronic Institutions*, pages 126–147. LNAI:1991. Springer-Verlag, 2001.
- [4] D. Greenwood, M. Lyell, A. Mallya, and H. Suguri. The IEEE FIPA approach to integrating software agents and web services. In *AAMAS '07*, pages 1–7. ACM, 2007.
- [5] J. Hubner, J. Sichman, and O. Boissier. S-Moise+: A middleware for developing organised multi-agent systems. In *Proc. OOP'06*, pages 64–78, 2006.
- [6] M. Sensoy, C. Pembe, H. Zirtiloglu, P. Yolum, and A. Bener. Experience-based service provider selection in agent-mediated e-commerce. *Eng. Appl. Artif. Intel.*, (3):325–335, 2007.

<sup>1</sup><http://www.dsic.upv.es/users/ia/sma/tools/Thomas>